



Màster universitari en **Formació del Professorat d'Educació Secundària  
Obligatòria i Batxillerat, Formació Professional i Ensenyament d'Idiomes**

## Treball de fi de màster

**Títol: Elaboració de material didàctic per introduir la programació d'ordinadors mitjançant pseudocodi i llenguatge de programació SMALL BASIC a Informàtica de 4rt d'ESO**

**Cognoms: TRIGO COBOS**

**Nom: FRANCESC XAVIER**

**Titulació: Màster en Formació del Professorat d'Educació Secundària Obligatòria i Batxillerat,  
Formació Professional i Ensenyament d'Idiomes**

**Especialitat: TECNOLOGIA**

**Director/a: ALFONSO RODRÍGUEZ GALÁN**

**Data de lectura: 26/06/2014**

## ÍNDIX

1. Introducció .....	3
2. Definició i context del problema .....	4
3. Objectius i competències treballades .....	5
4. La situació actual de la programació d'ordinadors .....	6
5. Descripció de la solució proposada .....	8
6. Material didàctic. Introducció I .....	9
7. Material didàctic. Introducció II .....	24
8. Conclusions .....	42
9. Bibliografia i referències .....	43

## 1. INTRODUCCIÓ

Durant l'etapa de l'ensenyament secundari obligatori i en concret a la matèria de les tecnologies, el currículum oficial introdueix una sèrie de conceptes i procediments que fa que els alumnes compreguin el funcionament, la seva utilització o manipulació de diversos objectes i processos tecnològics que es poden trobar a la vida quotidiana.

Entre aquests processos podem trobar, per exemple, la producció d'energia elèctrica. Durant la unitat didàctica de producció d'energia els alumnes aprenen el funcionament d'una central hidroelèctrica i per tant comprenen una forma de generació de l'electricitat. Posant aquest exemple, penso que l'ensenyament de la informàtica durant l'ESO queda una mica coixa, ja que als alumnes se'ls hi ensenya a utilitzar nombroses eines informàtiques però no aprenen o comprenen com funciona el programari que governa aquestes eines.

El programari són les instruccions que conté una eina o sistema informàtic per a que es porti a terme una determinada funció. Per tant el programari és present tant a petites aplicacions domèstiques per a fer una tasca en concret, com a tots els sistemes operatius molt coneguts o aplicacions corporatives que poden realitzar nombroses funcions. La majoria dels alumnes de 4rt d'ESO han nascut a la societat de la informació i la comunicació, i per tant l'ús i la presència de les tecnologies digitals és un fet habitual tant a l'escola com a les seves cases. Aquests alumnes han après a utilitzar ordinadors, telèfons, tabletas digitals quasi sense que ningú els hi hagi explicat. Però, ¿realment coneixen com està elaborat o com funciona el programari que controla totes les funcions que poden realitzar amb aquestes eines? Segurament no.

Seguint amb el símil de la producció d'energia podríem dir que l'ordinador és la central hidroelèctrica que fa possible que es pugui generar electricitat i el programari són les instruccions que té la persona o sistema encarregat d'obrir les comportes de la presa quan ni ha massa aigua. Amb tot això vull dir que hauríem d'anar una mica més enllà a l'hora d'ensenyar el funcionament de les eines informàtiques i desgranar una mica el seu funcionament intern mitjançant una petita introducció a la programació d'ordinadors.

He pensat que el moment adequat per a introduir la programació d'ordinadors hauria de ser a la matèria optativa d'Informàtica a 4rt d'ESO complementant així la matèria de Tecnologia del mateix curs on ja es fa una petita introducció a la programació i control de sistemes automàtics dins la unitat didàctica de Control i Automatització, i la resta de continguts del currículum de la optativa d'Informàtica.

Aquests alumnes escullen aquestes matèries per què comencen a dirigir els seus estudis cap a un batxillerat tecnològic que després pot encaminar-se cap a un grau universitari d'enginyeria, informàtica o un grau mig relacionat amb la mateixa temàtica on la programació d'ordinadors és també present. Per tant, he dissenyat aquest material didàctic tant per professors, com per alumnes, que té com a objectiu la realització d'una petita introducció a la programació d'ordinadors primer de tot mitjançant un llenguatge de programació de tipus pseudocodi[1].

El pseudocodi o fals llenguatge de programació té com a objectiu representar la solució a un problema de la forma més detallada possible el més semblant a un llenguatge de programació real per tant és una base que pot servir als alumnes per després aprendre a programar en la majoria dels llenguatges de programació. En el nostre cas, aquest material es complementa amb una introducció i unes petites activitats d'aplicació real dels casos vistos en pseudocodi utilitzant el llenguatge SMALL BASIC[2].

## 2. DEFINICIÓ I CONTEXT DEL PROBLEMA

Durant la meua estada de pràctiques a l'institut vaig poder parlar diverses vegades amb el cap del departament de Tecnologia que a la vegada era el responsable de la matèria Informàtica de 4rt d'ESO. Em va comentar diverses vegades que trobava a faltar una petita introducció a la programació d'ordinadors dins de l'optativa d'Informàtica de 4rt d'ESO.

Em comentava que ja fa molts anys que és el responsable de fer aquesta matèria i que cada curs que passa troba més a faltar una unitat didàctica que introdueixi la programació d'ordinadors en aquesta matèria. Des del seu punt de vista, molts dels alumnes que escullen aquesta matèria troben a faltar algun punt del currículum que sigui creatiu i que vagi més enllà del tractament de la imatge.

Debatent aquest tema vam arribar a la conclusió de que el currículum de les tecnologies i informàtica a l'ESO encamina els alumnes a que puguin seguir els seus estudis més enllà del batxillerat escollint una enginyeria o un cicle relacionat amb la tecnologia. Molts d'aquests estudis superiors tenen assignatures relacionades amb la programació d'ordinadors, és a dir, tenen assignatures on hauran de fer servir un llenguatge de programació i fer els seus propis programes. Amb això vull destacar que els alumnes que no hagin fet programació d'ordinadors de forma autodidacta o a un centre d'estudis privat, i que quan comencin els seus estudis superiors, no hauran vist mai què és la programació d'ordinadors, si no es fa una introducció des dels nostres instituts.

Aquesta introducció a la programació d'ordinadors a banda de aprendre a fer petits programes també permetrà que els alumnes compreguin com estan construïdes i esbrinar tot el que hi ha darrera de moltes de les aplicacions informàtiques que actualment ells mateixos fan servir quasi cada dia de les seves vides, ja sigui per oci o per fer els propis treballs de l'escola.

De totes aquestes converses vaig tenir la idea de fer aquest treball final de màster dedicat a la introducció de la programació d'ordinadors, tot aprofitant els meus coneixements en aquest àmbit. La meua experiència em diu que per aprendre a programar ordinadors és bàsic que els alumnes adquireixin primerament una base de la lògica de construcció de programes que sigui independent de qualsevol llenguatge de programació dels que existeixen actualment al mercat.

Aquesta base que anomenaré "pseudocodi" ens permetrà introduir la programació d'una forma fàcil i comprensible per l'alumnat. Una vegada l'alumne ha adquirit una petita base de programació és el moment de començar a introduir la programació directament amb un llenguatge de tipus comercial on podran veure de primera mà els resultats als problemes plantejats i on hauran fet el seu propi programa[1].

La programació d'ordinadors requereix un cert nivell de lògica i abstracció que els alumnes de 4r d'ESO ja haurien de tenir adquirit. A més a més, molts dels llenguatges de programació utilitzen instruccions que majoritàriament utilitzen paraules en anglès que els mateixos alumnes d'aquesta etapa ja poden comprendre perfectament.

Al final el que es vol aconseguir amb la introducció de la programació d'ordinadors a les aules de l'ESO és que els alumnes facin un canvi de rol i passin a ser creadors actius i controladors de la tecnologia enlloc de ser usuaris passius d'eines informàtiques i tecnològiques. La programació es útil per tenir un pensament ordenat i metòdic, i epistemològicament és molt semblant a qualsevol mètode de projectes.

### 3. OBJECTIUS I COMPETÈNCIES TREBALLADES

En conseqüència, donat tot l'exposat fins ara, atès el potencial existent respecte al professorat i material a la majoria de centres i considerant les aportacions que jo pugui realitzar, els objectius generals que em plantejo amb aquest TFM són:

1. DONAR resposta a la inquietud generada pel responsable de la matèria d'Informàtica a 4rt d'ESO del centre on vaig fer el pràcticum per tenir un material per treballar a l'aula.
2. PROPOSAR activitats en format de material didàctic per a poder dur a terme a l'aula d'informàtica de qualsevol centre.
3. ADJUNTAR una pauta de recomanacions i solucions a les activitats per professorat no expert en programació.
4. POSAR a prova, jo mateix, aquest material amb els alumnes del centre on feia el pràcticum.

#### Competències treballades

Els continguts de la solució proposada a la introducció a la programació d'ordinadors permet contribuir a l'adquisició, per part dels alumnes de 4rt d'ESO, d'una sèrie de competències bàsiques que s'emmarquen dins el currículum de l'educació secundària obligatòria [3].

Referent a la dimensió comunicativa, la programació d'ordinadors permet desenvolupar la competència comunicativa audiovisual mitjançant la possibilitat d'expressar la majoria de missatges amb la utilització d'un programa informàtic que els mateixos alumnes poden crear i personalitzar al seu gust. Dins de la mateixa dimensió comunicativa la programació també permet desenvolupar la competència artística i cultural. Avui en dia els programes informàtics permeten realitzar veritables obres d'art digitals, i per tant donat el cas, un programa fet pels mateixos alumnes permetria comunicar un missatge de forma artística.

Dins la dimensió metodològica podem trobar al currículum tres competències bàsiques on la programació d'ordinadors pot ajudar considerablement a la seva adquisició. La competència de tractament de la informació i la competència digital són clarament dues competències on la programació pot contribuir de forma molt important ja que com he comentant abans la programació d'ordinadors s'ha tornat imprescindible per comprendre el funcionament de les nombroses aplicacions informàtiques que s'utilitzen actualment. La programació també pot contribuir a l'adquisició de la competència matemàtica mitjançant l'ús de la lògica imprescindible per a realitzar qualsevol programa informàtic. Per últim, dins d'aquesta mateixa dimensió la programació pot contribuir a desenvolupar la competència d'aprendre a aprendre, despertant l'interès per descobrir els seus límits i augmentar la iniciativa dels propis alumnes per resoldre qualsevol problema plantejat mitjançant la programació d'un programa de pròpia creació.

El desenvolupament de la competència d'autonomia i iniciativa dins la dimensió personal és una altre de les competències on la programació d'ordinadors pot contribuir de forma considerada. Per a la realització d'un programa informàtic no existeix una fórmula exacta. Els alumnes poden utilitzar nombroses formes diferents de trobar la solució a un problema plantejat. La seva autonomia els permetrà trobar el camí i identificar quin és el camí més òptim o adequat per programar eficaçment la solució.

Per últim, ens fixarem en les competències de la dimensió de conviure i habitar el món. Amb la programació d'ordinadors també podem millorar les competències de coneixement i interacció amb el món físic, i la competència social i ciutadana. És necessari que els alumnes tinguin una

petita noció del funcionament del programari que governa a majoria d'eines digitals que utilitzem a la nostra vida quotidiana així com poder contribuir mitjançant la utilització de nous programes informàtics a la millora de diversos problemes o dificultats socials.

Resumint, podem veure que amb l'aprenentatge de la programació d'ordinadors podem contribuir a l'adquisició de totes les competències bàsiques que s'especifiquen al currículum de l'ESO. Per tant considero que aquesta introducció a la programació d'ordinadors pot ser una eina molt útil per assolir bona part dels objectius finals d'aquesta etapa educativa.

#### **4. LA SITUACIÓ ACTUAL DE LA PROGRAMACIÓ D'ORDINADORS**

##### **Què diuen els experts?**

Al 1996 la UNESCO va publicar un informe realitzat per experts en tecnologies de la informació on s'especificava un complet programa escolar per l'ensenyament de la informàtica a l'escola secundària. Aquest programa inclou totes les competències necessàries per a preparar els futurs adults i treballadors per a una futura societat totalment informatitzada. Aquest informe també contenia un apartat que introdueix la programació d'ordinadors dins del currículum de la matèria d'informàtica [4].

Quasi vint anys després, aquestes prediccions dels experts de l'informe de la UNESCO que anaven dirigides a totes les persones responsables d'elaborar programes escolars, han sobrepassat les expectatives i les tecnologies de la informació s'han convertit en un dels pilars bàsics de la societat actual.

Israel o Finlàndia són països pioners en introduir la programació als seus currículums oficials tant a la secundària com a la primària. Segons paraules del ministre d'educació finès: "Els nens d'avui estan creixent com nadius digital i fan servir la tecnologia abans d'aprendre a caminar, per tant la programació és en part culpable d'això".[5]

Per altre banda, Estònia va començar la seva aposta per introduir la programació a les escoles primàries al 2012 mitjançant un programa anomenat PROGETIIGER y que destaca en una sola frase la responsable del projecte: "Volem que els nostres nens no siguin només usuaris de software, sinó que es converteixin en els creadors de les noves tecnologies" [6].

El professor americà *Mitchel Resnick*, premi nacional de la ciència, premi McGraw d'educació i director dels grups d'aprenentatge de l'Institut Tecnològic de Massachusetts va ser un dels components que va desenvolupar el llenguatge de programació SCRATCH i que ara és molt utilitzat a nombrosos centres educatius de tot el món, per introduir els alumnes al món de la programació informàtica. A una de les seves conferències titulada "Ensenyem als nens a codificar" realitza una similitud entre aprendre a llegir i aprendre a programar, en base a que la capacitat d'aprendre a llegir ens facilita aprendre altres coses i per tant aprendre a programar no ha de servir solament per crear programadors, sinó que ha de ser la porta per accedir a un nou nivell de capacitat d'aprenentatge [7].

Al curs 2013-2014 el Regne Unit ha fet l'aposta per introduir la informàtica (inclosa la programació d'ordinadors), com a matèria troncal dins del seu batxillerat científic. Segons paraules del portaveu d'Educació del Govern Britànic: "Això significarà que milions de nens que estan aprenen a escriure codi informàtic passaran a ser creadors actius i controladors de la tecnologia en lloc de ser simples usuaris passius" [8].

A l'octubre de 2010 el professor *Antoine Petit*, subdirector general de l'Institut Públic d'Investigació i Tecnologia de França va publicar un informe a la revista INTERSTICES on destaca la seva preocupació per la manca d'alfabetització informàtica dels alumnes francesos quan acaben la secundària.

L' informe de Petit destaca que és necessari que els alumnes comprenguin una sèrie de conceptes relacionats amb els algorismes i la programació d'ordinadors que els serà molt útil per formar part de la nova societat digital global que impera a la economia de la majoria dels països. Ja al curs 2012-2013 les assignatures d'informàtica del currículum francès de secundària comencen a introduir la programació d'ordinadors com una eina no per preparar experts programadors, sinó com a eina per adquirir uns coneixements bàsics per a desenvolupar habilitats informàtiques com els algorismes i llenguatges de programació entre d'altres [9].

Per tant, sembla ser que aquesta disciplina informàtica comença a guanyar adeptes entre els responsables d'educació de països d'economies molt importants. Han valorat els avantatges que té ensenyar els nostres alumnes a programar ordinadors.

### **La informàtica de 4rt d'ESO a Catalunya**

Al nostre país, la matèria optativa d'Informàtica a 4rt d'ESO està programada per realitzar-se en 105 hores de dedicació. Els continguts d'aquesta matèria estan repartits en tres blocs anomenats de la següent forma: Creacions Multimèdia, Publicació i difusió de continguts i Eines per a la comunicació.

Sense cap dubte la programació d'ordinadors es podria integrar tant a la primera part de creacions multimèdia com a la segona de publicació i difusió, ja que podem realitzar qualsevol creació i publicació d'aquest tipus amb un programa dissenyat per nosaltres mateixos.

Per tant amb la introducció de la programació dins d'aquest curs els alumnes adquiriran una base de programació que els hi serà molt útil, per exemple, per la unitat didàctica de Control i Programació d'automatismes que es trobaran a la matèria de Tecnologia Industrial I del primer de batxillerat tecnològic.

## 5. DESCRIPCIÓ DE LA SOLUCIÓ PROPOSADA

La proposta d'aquest TFM consisteix en una sèrie de materials didàctics que permetran al professor fer una introducció a la programació d'ordinadors mitjançant diferents eines i exemples que ajudaran l'alumne a comprendre fàcilment aquest nou coneixement. Aquests materials estan dissenyats per una durada total de 12 sessions d'una hora cadascuna que correspondrien a aproximadament a un 12% del total d'hores programades a tot un curs de l'optativa de Tecnologia de 4rt d'ESO.

Aquest material l'he dividit en dues parts. Una primera part que serà una introducció bàsica a la programació i una segona part on es farà una introducció més completa a la programació informàtica amb una sèrie d'exercicis una mica més complexos.

Cal destacar que abans d'aplicar aquest material els alumnes ja hauran treballat a cursos anteriors les característiques de funcionament d'un ordinador i quines parts bàsiques el componen.

A partir de tot això he dissenyat una temporització d'aquest material de la següent forma:

### Primera Part: Introducció I

Sessió	Activitat 1	Activitat 2
1	Conceptes bàsics	
2	Fases per a la creació d'un programa	
3	El pseudocodi. 1ª Part	
4	Què és SMALL BASIC	
5	El pseudocodi. 2ª Part	El nostre primer programa en Small Basic
6	El pseudocodi 3ª Part	Activitat final en Small Basic

### Segona Part: Introducció II

Sessió	Activitat 1	Activitat 2
7	Estructures de control en pseudocodi	Estructures de control en Small Basic
8	Estructures repetitives en pseudocodi	Estructures repetitives en Small Basic
9	El mode gràfic a SMALL BASIC 1ª part	
10	El mode gràfic a SMALL BASIC 2ª part	
11 i 12	Activitat Final	



## 6. MATERIAL DIDÀCTIC. Introducció I.

### Sessió 1. Introducció I. Conceptes bàsics.

#### **Nota:**

Durant aquesta primera sessió els alumnes hauran de comprendre els conceptes bàsics de què és programar, per a que serveixen els programes informàtics, què és un algoritme, què és el pseudocodi i què és un llenguatge de programació.

- ✓ *És important que els alumnes sàpiguen què és un programa informàtic i tinguin clara la diferència entre pseudocodi i llenguatge de programació*

#### **Què és la programació d'ordinadors ?**

És l'acte de crear un programa, que és un conjunt d'instruccions que un ordinador pot executar, que està escrit en un llenguatge de programació i que té com a objectiu resoldre un problema en base a uns requeriments específics.

#### **Per a què serveixen els programes informàtics ?**

Els programes informàtics serveixen per crear aplicacions que resolen o ajuden a resoldre diversos problemes de la vida real. Aquestes solucions poden ser de diferents tipus:

- Científics
- Comercials
- Administratius
- Lúdiques
- Altres

#### **Què és un algoritme ?**

Un algoritme és l'execució planificada d'una sèrie d'accions en un ordre específic que serveix per a trobar la solució a un problema plantejat. És a dir, un procediment per resoldre un problema que conté les accions a executar-se en un ordre determinat.

Un exemple d'algoritme per a arribar a l'escola seria:

- Problema plantejat: Arribar a l'escola
- Accions i ordre:
  1. Sortir del llit
  2. Treure's el pijama
  3. Dutxar-se
  4. Vestir-se
  5. Esmorzar
  6. Caminar fins a l'escola

#### **Proposta d'exercici:**

Realitzar l'algoritme per enviar un missatge amb el correu electrònic del teu ordinador

## Què és un pseudocodi ?

Un pseudocodi és un llenguatge que s'assembla a un llenguatge de programació, però que no segueix unes regles estrictes ni de gramàtica ni de sintaxis. És a dir, un llenguatge que permet expressar algorismes (solucions a problemes) d'una manera natural e independent de qualsevol llenguatge de programació[1].

Un exemple de pseudocodi que ensenyaria en pantalla la suma de 3 + 5 seria:

```
Inici
  A=3
  B=5
  Presenta_a_pantalla ( A+B )
Fi
```

## Què és un llenguatge de programació ?

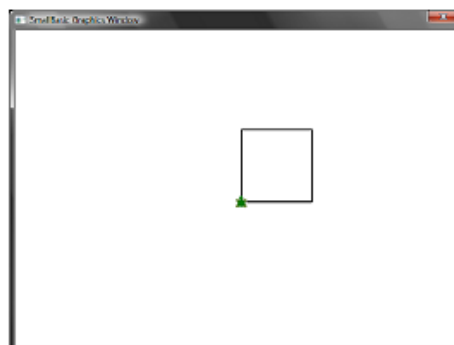
- És un conjunt de símbols, instruccions i enunciat que estan subjectes a unes regles i que permeten la creació d'un programa informàtic.
- En un principi es feien servir llenguatges de programació anomenats de baix nivell i que utilitzaven una codificació molt propera a l'ordinador però més difícil de utilitzar pel programador.
- Actualment es fan servir llenguatges de programació d'alt nivell que són llenguatges més semblants a la llengua humana que utilitzen frases o paraules normalment en anglès i que permeten fer creacions o modificacions amb més facilitat.

Per posar un exemple senzill de programa, si volem dibuixar un quadrat a la pantalla de l'ordinador les instruccions que haurem d'escriure amb un llenguatge de programació serien:

### Nota:

Amb aquest petit programa li comuniquem a l'ordinador que repetirem 4 vegades una combinació de dues instruccions que permeten dibuixar el quadrat. La primera dibuixa una línia de 100 píxels i la segona gira el cursor 90 graus a la dreta per poder dibuixar la següent línia, i així quatre vegades s'aconsegueix l'efecte de dibuixar un quadrat.

```
For i = 1 To 4
  Turtle.Move(100)
  Turtle.TurnRight()
EndFor
```



## Sessió 2. Introducció I. Fases per a la creació d'un programa.

### Nota:

En aquesta sessió els alumnes aprendran les fases que són necessàries per a la creació de qualsevol programa informàtic. A partir d'un problema pensat per ells mateixos hauran de fer totes les passes necessàries per construir el seu propi algoritme.

- ✓ És molt important que els alumnes s'acostumin a identificar amb el màxim de detall totes les etapes per solucionar un problema plantejat.

### 1. Definició del problema

En aquesta fase se'ns presenta l'enunciat del problema el qual requerirà una definició clara i precisa.

### 2. Anàlisi del problema

Una vegada que tenim clar el que es demana a l'enunciat del problema s'ha de definir:

- Les dades d'entrada (dades necessàries per començar a resoldre el problema)
- Les dades de sortida (dades o informació que es vol produir)
- Mètodes i fórmules necessàries per processar les dades

### 3. Disseny de l'algoritme o del pseudocodi

Les característiques d'un bon algoritme són:

- Ha de tenir un punt d'inici particular
- Ha de ser definit i no ha de permetre dobles interpretacions
- Ha de ser genèric i que permeti suportar la majoria de variants que pugui tenir la definició del problema
- Ha de ser finit en grandària i temps d'execució

### 4. Codificació

La codificació és la operació d'escriure la solució del problema d'acord a la lògica de l'algoritme o del pseudocodi amb una sèrie d'instruccions detallades en un codi o llenguatge de programació que l'ordinador pot reconèixer.

### 5. Prova i depuració

La prova i depuració és el procés d'identificar i eliminar els errors que s'hagin pogut fet a l'hora de codificar la solució del problema.

## 6. Documentació

Moltes vegades un programa es escriu per una persona, però utilitzat per altra persona per tant, és necessari que hi hagi una documentació que ajudi a comprendre i a utilitzar el programa per un usuari i documentació que serveixi per facilitar futures modificacions o manteniments per part d'altres programadors.

## 7. Manteniment

És la fase en que una vegada acabat el programa, quan es detecta que és necessari fer qualsevol canvi, ajust o complement del programa per a que pugui seguir treballant correctament.

### Proposta d'exercici:

Demanar als alumnes que en grups de 4 components facin la definició d'un problema plantejat per ells mateixos, facin l'anàlisi del problema i el disseny de l'algoritme.

## Sessió 3. Introducció I. El pseudocodi 1ª part.

### Nota:

El pseudocodi és una eina molt utilitzada en el disseny de programes que permet als programadors expressar els seus pensaments d'una forma clara i ordenada amb un llenguatge natural i sense ambigüitat.

- ✓ L'objectiu principal del pseudocodi és representar la solució a un problema de la forma més detallada possible i el més semblant a la sintaxi que es farà servir posteriorment per codificar en un llenguatge de programació.

## Els tipus de dades

### Tipus de dades bàsics

*Enter:* És un conjunt finit de números enters  
Exemples: 6 15 346 11234

*Decimal:* És un conjunt finit de números amb decimals.  
Exemples: 1,23 234,56 56743,89

**Caràcter:** Conjunt ordenat de caràcters (Lletra u número). Els caràcters es representen entre cometes simples.

Exemples: '0'..'9', 'a'..'z' i 'A'..'Z' .

Un espai en blanc també és un caràcter: ' '

**Cadena:** Conjunt de caràcters limitats entre doble cometa.

Exemples: "HOLA", "casa", "Ruta66"

**Lògic:** Conjunt format pels valors TRUE o FALSE

Exemples: Es representen per 'T' o 'F' o '1' o '0'

### Tipus de dades derivats

**Constant:** Són dades les quals el seu valor no canvia mai durant l'execució d'un programa. Per tant podem tenir tants tipus de constants com tipus de dades bàsics.

Constant Entera: 30  
Constant Decimal: 3,1416  
Constant Caràcter: 'A'  
Constant Cadena: '4rt ESO'  
Constant Lògica: 1

**Variable:** Són dades les quals el seu valor pot canviar durant l'execució d'un programa. S'identifiquen per un nom, anomenat identificador i pel seu tipus bàsic. Els identificadors ha de complir unes regles i es representa per una seqüència de lletres de la **a** a la **z** minúscula; de la **A** a la **Z** majúscula i dígit del **0** al **9**. Sempre ha de començar per una lletra, mai per un dígit.

Exemples: Mes, Dia, Any, DATA, Dia1, Dia2, DATA\_INICI, Data\_fi

### Els operadors

Un operador és un caràcter o grup de caràcters que actua sobre una o més variables per a realitzar una determinada operació amb un determinat resultat. Alguns exemples de operadors són: (+), (-), (\*), (/) ....

#### Operadors aritmètics

Suma:	+	25+25 = 50
Resta:	-	12-2 = 10
Multiplicació:	*	4*5 = 20
Divisió:	/	20/5 = 4
Mòdul:	%	10%3 = 1 (Resto de la divisió 10/3)

## Operadors d'assignació

Un operador d'assignació atribueix a una variable un valor o un resultat d'una operació. L'operador d'assignació més utilitzat és: =

Exemple:

Si vols assignar el valor de la teva edat a una variable es faria així: *Edat* = 16

Si vols assignar el teu nom a una variable ho farem així: *Nom* = "David"

## Operadors relacionals

Els operadors relacionals permeten estudiar si unes variables compleixen unes condicions determinades.

Per exemple, si volem comparar dues variables per veure quina té un valor més gran utilitzarem l'operador relacional > d'aquesta forma A>B

A un programa si es compleix la condició es TRUE i si no la compleix serà FALSE.

Els operadors relacionals més utilitzats són:

Igual que:	==
Més gran que:	>
Menor que:	<
Més gran o igual que:	>=
Menor o igual que:	<=
Diferent que:	!=

Alguns exemples d'utilització d'operadors relacionals poden ser:

2 = 1	Això ens dona FALSE (2 no és igual a 1)
1 = 1	Això ens dona TRUE (1 és igual a 1)
2 <= 2	Això ens dona TRUE (2 es igual o menor a 2)
1 != 1	Això ens dona FALSE (1 no és diferent de 1)

## Operadors lògics

Els operadors lògics són operadors que permeten combinar els resultats dels operadors relacionals, comprovant que es compleixen unes condicions necessàries.

Podem distingir els operadors lògics següents:

Operador AND  
Operador OR  
Operador NOT

Els operadors lògics AND i OR es poden combinar entre parèntesi, com per exemple:

```
(2=1) OR (-1=-1)           // el resultat és 1 o TRUE
(2=2) AND (3=-1)           // el resultat és 0 o FALSE
((2=2) AND (3=3)) OR (4=0)  // el resultat és 1 o TRUE
((6=6) OR (8=0)) AND ((5=5) AND (3=2)) // el resultat és 0 o FALSE
((250) AND (330)) 250>0,330>0 // el resultat és 1 o TRUE
```

### Proposta d'exercici:

Quin resultat tindrien les següents operacions (TRUE o FALSE) :

- a)  $4 = 1$
- b)  $(3=2) \text{ OR } (-3 = -3)$
- c)  $9 = 9$
- d)  $(1=1) \text{ AND } (5= -3)$
- e)  $5 \leq 5$
- f)  $((0=1) \text{ OR } (2=2)) \text{ AND } (4=4)$
- g)  $8 \neq 8$
- h)  $((2=2) \text{ OR } (3=0)) \text{ AND } ((3=3) \text{ AND } (5=3))$

## Sessió 4. Introducció I. Què és SMALL BASIC?

### Nota:

SMALL BASIC és un llenguatge de programació dissenyat exclusivament per assolir els conceptes bàsics de la programació d'ordinadors. Coneixerem la seva estructura i com es treballa amb ell.

- ✓ En aquesta sessió és important que els alumnes identifiquin i es familiaritzin amb totes les parts que componen l'editor Small Basic.

### Introducció al llenguatge de programació SMALL BASIC

El llenguatge de programació SMALL BASIC és un llenguatge de la companyia americana Microsoft que té una interfície de treball molt senzilla i que està dissenyat per a fer la programació divertida i accessible per a tots els principiants.

L'entorn de treball no es complica d'entendre i de fer servir, i quan l'executem per primera vegada veurem una pantalla semblant a la següent figura.

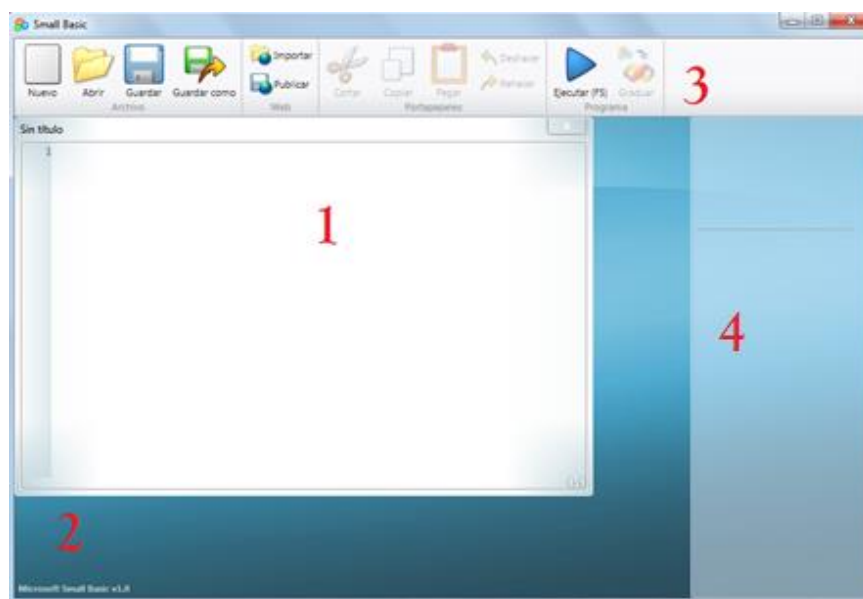


Figura 1 – L'entorn Small Basic

Aquest és l'entorn de Small Basic, on escriurem i executarem els nostres programes fets amb el llenguatge Small Basic. Com es pot veure a la figura 1, l'entorn conté diferents zones identificades amb el número en color vermell.

La zona de l' **Editor**, identificat amb el número 1 és el lloc on escriurem els nostres programes. A l'editor podrem obrir, modificar i gravar els nostres programes per a usos posteriors.

La **Barra d'eines**, identificada amb el número 2, s'utilitza per executar accions o comandaments que ens seran útils per fer els nostres programes.

La zona identificada amb el número 3, és el lloc on es veuran tots els editors, ja que es pot donar el cas de poder tenir més d'un editor a la vegada, és a dir, pots estar programant diferents programes a la vegada.

Per últim, la zona marcada amb el número 4 ens permet veure les diferents propietats de les instruccions que posem a l'editor. És com un tipus d'ajuda que ens facilitarà la programació.

### Formats de treball

El llenguatge de programació Small Basic ens permet treballar de dos formats diferents depenen els tipus de programa que volem realitzar i segons el que volem programar.

Small Basic ens permet treballar en format text, és a dir utilitzant la finestra d'execució de comandaments del sistema operatiu Windows per a veure el resultat de l'execució del nostre programa.



Un primer format de tipus text que tindria una visió en pantalla com la següent figura:

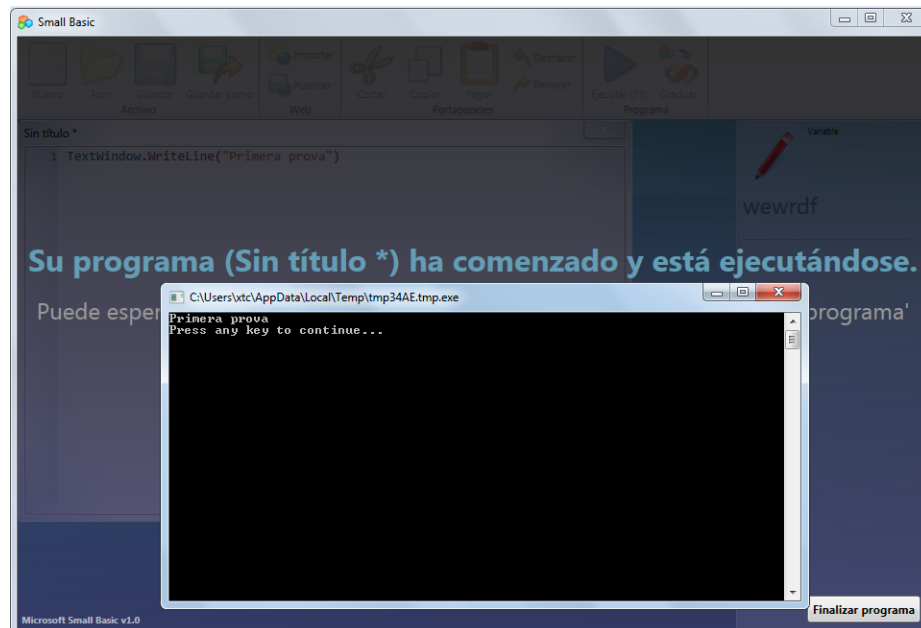


Figura 2 – Small Basic: Execució en format text

El segon format que ens permet Small Basic és de tipus gràfic i es veuria en pantalla com la següent figura:

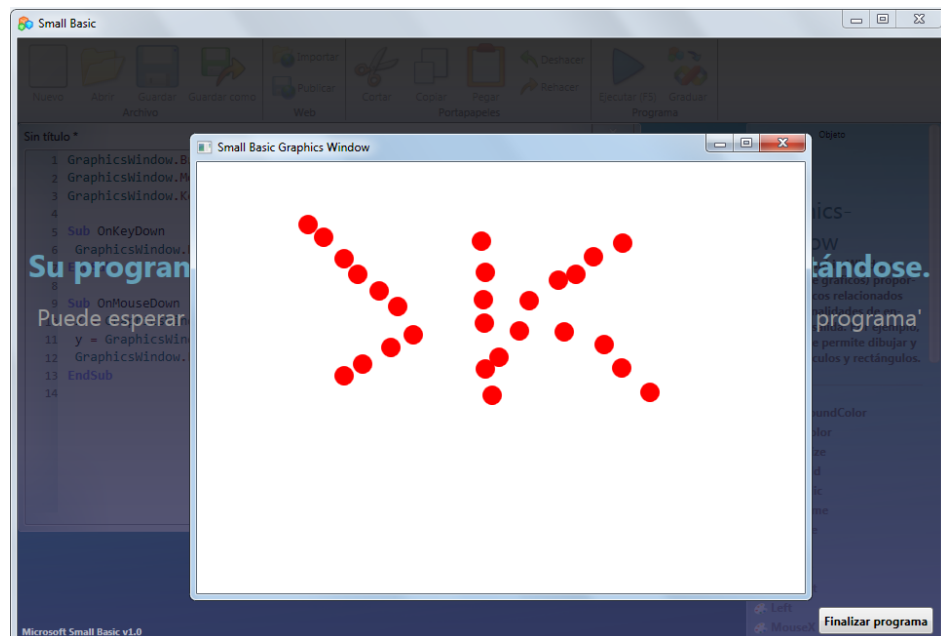


Figura 3 – Small Basic: Execució en format gràfic

## L'editor SMALL BASIC

L'editor que incorpora aquest llenguatge té diverses funcionalitats que ens ajudaran a programar fàcilment. Com es pot veure a la figura 4, quan escrivim una instrucció a l'editor podem veure les possibles accions que podem realitzar amb aquesta instrucció mitjançant una petita finestra emergent.

Al mateix temps a la àrea d'ajuda de la part dreta podem observar com és la sintaxi d'aquesta instrucció que ens ajudarà a escriure la instrucció evitant cometre errors.

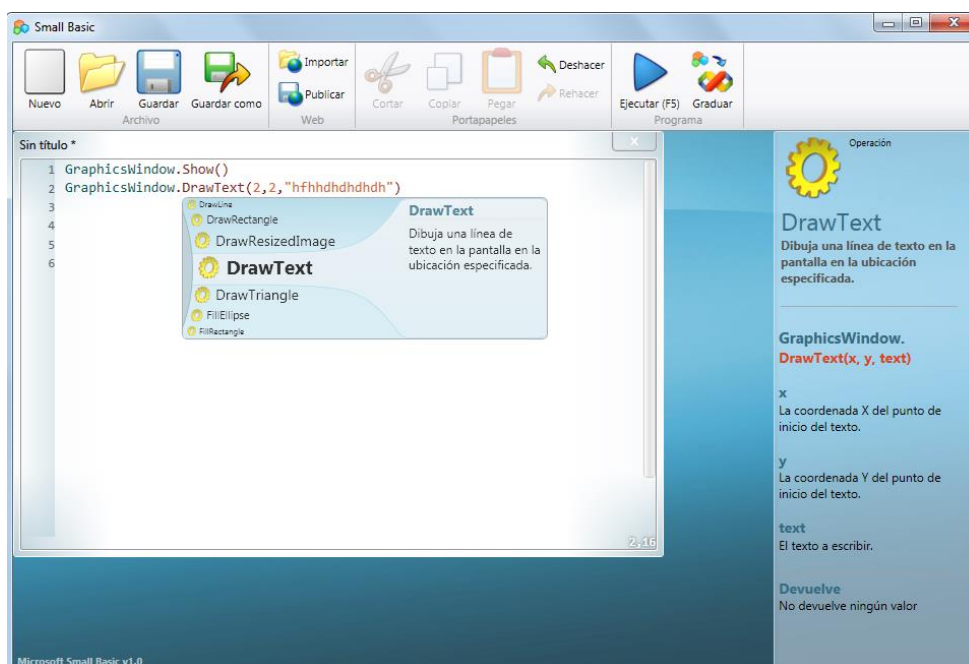


Figura 4 – L'editor SMALL BASIC

### Nota:

Al final de la sessió podem deixar uns minuts per a que els alumnes investiguin de forma autònoma tot el que s'ha après abans i preguntin els dubtes que puguin sorgir.

- ✓ És important que coneguin molt bé l'editor SMALL BASIC ja que després ens facilitarà molt la feina per fer els primers programes.

## Sessió 5. Introducció I. El pseudocodi 2ª Part. El nostre primer programa

### Nota:

En aquesta sessió acabarem d'introduir els últims conceptes importants del pseudocodi i farem alguns exemples de programes a l'editor SMALL BASIC.

- ✓ En programació és molt important que els alumnes adquireixin un pensament abstracte que els serà molt útil per resoldre els problemes plantejats i que es codifiquin mitjançant la realització d'un programa informàtic.

### Les funcions

A la majoria de llenguatges de programació existeixen funcions que a partir d'un valor retornen un resultat determinat. Per exemple:

<u>Funció</u>	<u>Descripció</u>
<i>abs(x)</i>	Valor Absolut
<i>cos(x), sin(x)</i>	Cosinus, Sinus
<i>quadrat(x)</i>	x <sup>2</sup>
<i>sencera(x)</i>	Part sencera
<i>ln(x)</i>	Logaritme neperià
<i>log(x)</i>	Logaritme en base 10
<i>arrel(x)</i>	Arrel quadrada
<i>arrodonir(x)</i>	Arrodonir número

Per exemple, si volem calcular la part sencera del número PI:

```
PI = 3,1416
PI_sencer = sencera(3,1416) o PI_sencer = sencera(PI)

// Ambdós casos ens donarà com a resultat 3
```

### Les expressions

Una expressió és una combinació de variables i/o constants i operadors. Per exemple 1+3 és una expressió amb dos operands (1,3) i un operador (+) que dona un resultat (4).

Per exemple si volem escriure aquesta equació farem servir l'expressió:

$$\sqrt{4ab^2}$$

```
arrel(4*a*quadrat(b))
```

### Proposta d'exercici:

Expressar en format pseudocodi la fórmula per calcular l'àrea d'una circumferència.

## Instruccions per a l'entrada i sortida de dades

### Sortida

La instrucció és `Escriure()` i presenta al dispositiu de sortida (pantalla o impressora) les expressions, valors, variables, cadenes que posem entre parèntesi. Per exemple en pseudocodi escriurem:

```
Escriure("El text és ", valor) // Veurem a pantalla el valor
                                // Si valor="HOLA" veurem a pantalla HOLA
```

En llenguatge Small Basic escriurem el següent codi a l'editor i polsarem el botó executar o F5:

```
text = "HOLA"
TextWindow.WriteLine("El text és " + valor)
```

### Entrada

La instrucció per recollir dades des del dispositiu d'entrada (normalment el teclat) utilitzarem `Llegir()`. Aquesta instrucció emmagatzemarà el valor entrat per teclat a la variable que li assignem. Per exemple en pseudocodi:

```
Llegir(X)   'Entrarem per teclat el valor que assignarem a la variable X
X=Llegir()
Escriure(X) 'Escriurem a pantalla el valor de X
```

A l'editor Small Basic escriurem:

```
TextWindow.Write("Introdueix el valor de X: ")
X = TextWindow.ReadNumber()
TextWindow.WriteLine("El valor és: " + X)
```

## El nostre primer programa

Una vegada ja hem introduït aquests conceptes podem passar a construir el nostre primer programa en SMALL BASIC.

### Proposta d'exercici:

Escriu en codi SMALL BASIC que et preguntui el teu nom i la teva edat, i que després el presenti en pantalla de la forma "Em dic XXX i tinc XX anys".

### Solució:

```
TextWindow.Write("Com et dius ? ")
nom = TextWindow.Read()
TextWindow.Write("Quina és la teva edat ? ")
X = TextWindow.ReadNumber()
TextWindow.WriteLine("Em dic " + nom + " i tinc " + X + " anys")
```

## Sessió 6. Introducció I. El pseudocodi 3ª part. Activitat final.

### Nota:

A aquesta darrera sessió de la primera part de la introducció a la programació els alumnes obtindran unes pautes molt clares per construir un programa i faran un programa utilitzant operacions matemàtiques.

- ✓ Amb la programació informàtica els alumnes aprenen a afrontar el plantejament d'un problema i aprenen a organitzar la solució mitjançant una seqüència lògica de passos, així com la formulació de decisions.

Coses a tenir en compte per escriure el nostre programa en pseudocodi:

- Tots els programes comencen amb la paraula **Inici** seguit d'un grup d'instruccions.
- Un grup d'instruccions o sentències es posaran entre claus { }.
- Es declararan com a norma totes les variables a l'inici del programa.
- Els comentaris comencen per '

Suposem que volem fer un programa que calculi l'àrea d'un triangle. Ja sabem que la fórmula per calcular l'àrea d'un triangle és:  $A = b \times h / 2$ . Per tant comencem a fer el nostre primer programa. Cal tenir en compte que poden haver moltes solucions diferents a l'hora de programar. En pseudocodi escriuríem:

```
Inici
{
  enter A,b,h;          ' Primer declarem les variables que farem servir
  Escriure("Entra el valor de la base i l'alçada"); ' Missatge a pantalla
  Llegir(b);            ' Entrem per teclat la base
  Llegir(h);            ' Entrem per teclat l'alçada
  A = b*h/2;            ' Calculem l'àrea
  Escriure("L'àrea és:" , A); ' Presentem el resultat a pantalla
}
```

En Small Basic escriurem a l'editor:

```
TextWindow.Write("Introdueix el valor de la base: ")
b = TextWindow.ReadNumber()
TextWindow.Write("Introdueix el valor de l'alçada: ")
h = TextWindow.ReadNumber()
A = b*h/2
TextWindow.WriteLine("L'àrea és: " + A)
```

Ara farem un programa en pseudocodi que calculi l'àrea de tres cercles. Entrarem per teclat el valor del R corresponent al radi del primer cercle i calcularem l'àrea dels tres cercles tenint en compte que el segon cercle té un radi de 2R i el tercer cercle un radi de 3R.

```
Inici
{
  decimal A1,A2,A3, R;
  decimal PI = 3,1416;
  Escriure ("Entra el valor del primer radi:");
  Llegir(R);
  A1 = 2*PI*R*R;
  A2 = 2*PI*2*R*R;
  A3 = 2*PI*3*R*R;
  Escriure ("L'àrea del primer cercle és:", A1);
  Escriure ("L'àrea del segon cercle és:", A2);
  Escriure ("L'àrea del tercer cercle és:", A3);
}
```

Ara implementarem l'exemple anterior a l'editor de Small Basic per a comprovar el seu funcionament per pantalla:

```
PI = 3.1416
TextWindow.Write("Introdueix el valor del primer radi: ")
radi1=TextWindow.ReadNumber()
Area1 = 2*PI*(radi1*radi1)
Area2 = 2*PI*2*(radi1*radi1)
Area3 = 2*PI*3*(radi1*radi1)
TextWindow.WriteLine("El valor de l'àrea 1 és: " + Area1)
TextWindow.WriteLine("El valor de l'àrea 2 és: " + Area2)
TextWindow.WriteLine("El valor de l'àrea 3 és: " + Area3)
```

### Proposta d'Activitat Final de la primera part:

Escriu primer en pseudocodi i després en llenguatge SMALL BASIC un programa que et pregunti una velocitat i un temps referent a un viatge que hagi fet en avió i et calculi la distància recorreguda en kilòmetres.

#### Solució 1:

```
TextWindow.Write("Velocitat de l'avió (en km/h) ? ")
velocitat = TextWindow.ReadNumber()
TextWindow.Write("Temps del viatge (en hores) ? ")
temps = TextWindow.ReadNumber()
distancia = (velocitat * temps)
TextWindow.WriteLine("La distància és de " + distancia + " km")
```

Ara modifica el programa anterior per a que et pregunti la distància i el temps transcorregut, i et pregunti també el destí del viatge. Hauràs de escriure en pantalla la velocitat de l'avió (arrodonida sense decimals) i el destí.

#### Solució 2:

```
TextWindow.Write("Destí del viatge ? ")
desti = TextWindow.Read()
TextWindow.Write("Distància a destí (en km) ? ")
distancia = TextWindow.ReadNumber()
TextWindow.Write("Temps del viatge (en hores) ? ")
temps = TextWindow.ReadNumber()
velocitat = Math.Round(distancia / temps)
TextWindow.WriteLine("La velocitat per arribar a " + desti + " va ser de " +
velocitat + " km/h")
```

#### Nota:

Amb aquesta darrera sessió de la primera part els alumnes hauran fet una primera introducció a la programació d'ordinadors que els haurà permès desenvolupar el seu raonament lògic mitjançant la construcció de petits programes tant en pseudocodi com en llenguatge Small Basic.

## 6. MATERIAL DIDÀCTIC. Introducció II.

### Sessió 1. Introducció II. Estructures de control.

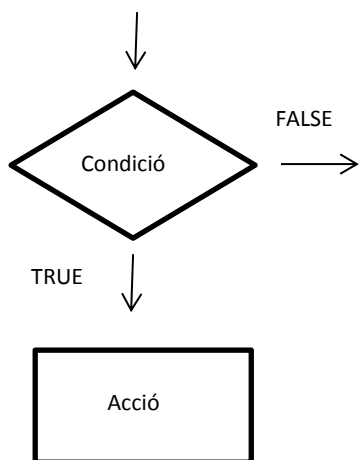
#### Nota:

La segona part d'aquest material comença amb una sessió dedicada a introduir conceptes més complexos però que són de molta importància per a la programació d'ordinadors com són les estructures de control. La programació de gestió més actual està basada en la metodologia de la programació estructurada, és a dir, passem d'escriure programes seqüencials tal i com hem fet a la primera part, a escriure programes amb estructures que permeten controlar el flux de decisions.

- ✓ Les estructures de control són la peça clau, i per tant s'ha de treballar molt de forma gràfica, que són més fàcil de comprendre, mitjançant organigrames o diagrames de flux. Les estructures de control permeten desenvolupar un pas més el raonament lògic dels alumnes.

A alguns programes l'evolució natural del mateix pot requerir d'unes estructures de control que faran variar l'execució segons es compleixin o no unes condicions determinades.

Una estructura condicional es representa de la següent forma:



En pseudocodi ho escriurem de la següent forma:

```
Si (condició) llavors <accio1> sinó <accio2>;  
Si (condició) llavors { <accio1>;  
                      <accio2>;  
                      ...  
                      }  
sinó <accio3>;
```

Un exemple en pseudocodi que et diu si ets major d'edat o no depenen de l'edat que entrem amb el teclat seria:

```
Inici  
{  
  enter Edat;  
  Escriure("Escriu la teva edat:");  
  Llegir(Edat);  
  Si (Edat>17) Llavors Escriure("Ets major d'edat");  
                Sinó Escriure("Ets menor d'edat");  
}
```



El mateix exemple codificat en Small Basic seria:

```
TextWindow.Write("Introdueix la teva edat: ")
edat=TextWindow.ReadNumber()
If edat>17 then
    TextWindow.WriteLine("Ets major d'edat")
Else
    TextWindow.WriteLine("Ets menor d'edat")
EndIf
```

Un altre exemple en pseudocodi que et diu si pots conduir un cotxe depenen de dues condicions entrades per teclat seria:

```
Inici
{
    enter Edat;
    caràcter Carnet;
    Escriure("Escriu la teva edat:");
    Llegir(Edat);
    Escriure("Tens carnet de conduir:");
    Llegir(Carnet);
    Si (Edat>17 && Carnet="S") Llavors Escriure("Pots conduir cotxe");
        Sinó Escriure("No pots conduir cotxe");
}
```

En Small Basic l'exemple anterior el codificarem de la següent forma:

```
TextWindow.Write("Introdueix la teva edat: ")
edat=TextWindow.ReadNumber()

TextWindow.Write("Tens carnet de conduir ? ")
carnet=TextWindow.Read()
If edat>17 and carnet = "S" then
    TextWindow.WriteLine("Pots conduir")
Else
    TextWindow.WriteLine("No pots conduir")
EndIf
```

Una altra forma diferent de fer el darrer exemple en Small Basic seria preguntant primer l'edat i només preguntar si té carnet en el cas de que sigui major d'edat.

```
TextWindow.Write("Introdueix la teva edat: ")

edat=TextWindow.ReadNumber()
If edat<18 then
    TextWindow.WriteLine("No pots conduir")
Else
    TextWindow.Write("Tens carnet de conduir ? ")
    carnet=TextWindow.Read()
    If carnet = "S" then
        TextWindow.WriteLine("Pots conduir")
    else
        TextWindow.WriteLine("No pots conduir")
    EndIf
EndIf
```

#### Proposta d'exercici:

Fes un programa en SMALL BASIC que et pregunti tres números i et presenti en pantalla el més gran de tots de la següent forma: "El número més gran és: XX"

#### Solució:

```
TextWindow.Write("Entra el primer número: ")
X = TextWindow.ReadNumber()
TextWindow.Write("Entra el segon número: ")
Y = TextWindow.ReadNumber()
TextWindow.Write("Entra el tercer número: ")
Z = TextWindow.ReadNumber()
If (X>Y) And (X>Z) then
    TextWindow.WriteLine("El número més gran és: " + X )
Else
    If X<Z then
        TextWindow.WriteLine("El número més gran és: " + Z )
    else
        TextWindow.WriteLine("El número més gran és: " + Y )
    EndIf
EndIf
```

## Sessió 2. Introducció II. Estructures repetitives.

### Nota:

En aquesta sessió introduïrem unes eines molt utilitzades en programació com són les estructures repetitives. Mitjançant aquestes estructures podem simplificar la codificació d'un programa ja que molt sovint ens permetran no repetir instruccions inútilment.

- ✓ Una vegada vistes les dues estructures més importants en programació, les estructures de control i les estructures repetitives els alumnes estaran preparats per poder programar la majoria de solucions a molts problemes quotidians.

### Estructura Des de .... fins

Una variable contindrà els valors el número de vegades que volem realitzar una seqüència de codi, per tant haurem de conèixer prèviament el número de vegades que es farà.

```
Des de i=0 fins i=10
{
    Escriure ( i )
}
```

‘ Aquest codi ens presentarà a pantalla els números  
‘ del 0 al 10

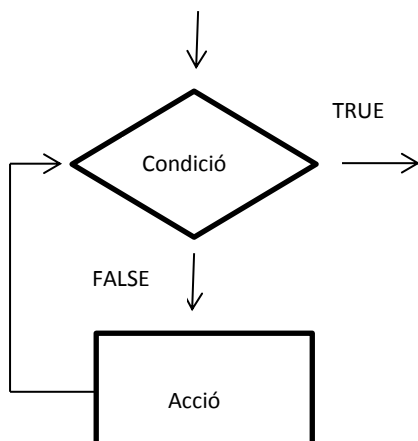
Ara representarem aquesta estructura es Small Basic:

```
For i=0 To 10
    TextWindow.WriteLine(i)
EndFor
```

### Estructura Mentre

Aquesta estructura repetirà una acció o grup d'accions mentre no es compleixi una determinada condició. En el moment que es compleixi la condició es sortirà del bucle.

Gràficament es representaria de la següent forma:



En pseudocodi l'exemple anterior seria:

Mentre (condició)	enter i;
{	i = 0;
accio1	Mentre (i<=10)
accio2	{
...	Escriure ( i )
}	i = i +1
	}

L'estructura MENTRE la representarem en Small Basic de la següent forma:

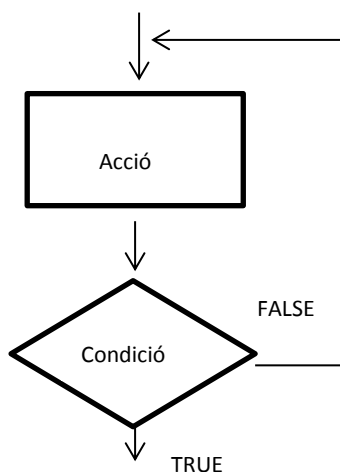
```

i=0
While i<=10
  TextWindow.WriteLine(i)
  i = i + 1
EndWhile
  
```

## Estructura Repetir

En aquesta estructura l'acció o grup d'accions es repetirà fins que es compleixi la condició i en el moment que es compleixi sortirà del bucle. Amb l'estructura REPETIR l'acció s'executa sempre, com a mínim una vegada, fins que es compleix la condició.

Gràficament es representaria de la forma següent:



En pseudocodi l'exemple anterior seria així:

<b>Repetir</b>	enter i
{	i = 0
accio1	Repetir
accio2	{
}	i = i +1
<b>Fins</b> (condició)	Escriure ( i )
	}
	Fins (i==10)

En Small Basic no existeix una estructura equivalent a repetir, però podem fer servir qualsevol de les altres estructures repetitives explicades anteriorment. Ara farem un exemple que combinarà una estructura repetitiva i una estructura de control per fer un petit programa que ens dirà si un número de l'1 al 20 es parell o imparell.

```
numero=0
While numero<=20
    reste = Math.Remainder(numero, 2)
    If reste = 0 then
        TextWindow.Write(numero)
        TextWindow.WriteLine(" És parell")
    else
        TextWindow.Write(numero)
        TextWindow.WriteLine(" És imparell")
    EndIf
    numero=numero+1
EndWhile
```

#### Proposta d'exercicis:

- Fes un programa en SMALL BASIC que presenti en pantalla tots els números de l'1 al 100.
- Fes un programa en SMALL BASIC que presenti en pantalla tots els números del 100 a l'1.
- Fes un programa en SMALL BASIC per a que faci la suma de tots els números IMPARELLS de l'1 al 100.
- Fes un programa en SMALL BASIC per a que faci la suma de tots els números PARELLS de l'1 al 100.

#### Solució:

- ```
For i=1 To 100
    TextWindow.WriteLine(i)
EndFor
```
- ```
For i=100 To 1
    TextWindow.WriteLine(i)
EndFor
```
- ```
Suma = 0
For i=1 To 100
    If math.Remainder(i,2) <> 0 then
        Suma = Suma + i
    EndIf
EndFor
TextWindow.WriteLine("Suma:" + Suma)
```
- ```
Suma = 0
For i=1 To 100
    If math.Remainder(i,2) = 0 then
        Suma = Suma + i
    EndIf
EndFor
TextWindow.WriteLine("Suma:" + Suma)
```

### Sessió 3. Introducció II. El mode gràfic a SMALL BASIC 1ª Part.

#### Nota:

En aquesta sessió farem una introducció al mode gràfic de Small Basic. El mode gràfic ens permetrà conèixer les eines disponibles d'aquest llenguatge de programació per poder treballar amb diferents elements gràfics com línies, figures, colors etc...

- ✓ El mode gràfic és un element molt motivador de cara a ensenyar programació d'ordinadors als alumnes ja que actualment quasi totes les aplicacions informàtiques que coneixem d'àmbit domèstic treballen sota un entorn gràfic al que els alumnes estan més acostumats a veure o treballar.

Fins ara em fet servir el mode text d'aquest llenguatge de programació, però Small Basic ens ofereix la possibilitat d'utilitzar el seu mode gràfic que ens permetrà dibuixar coses. La finestra negra que hem fet servir als exemples anteriors ara desapareix i es converteix en una finestra blanca.

La instrucció que posarem a l'editor i que ens permetrà veure la finestra gràfica de color blanc serà:

```
GraphicsWindow.Show()
```

#### Configuració de la finestra gràfica

A les finestres gràfiques podem posar un títol, es poden canviar les mides de la finestra i podem canviar el color de fons. A la finestra gràfica podem dibuixar figures, text i fins i tot imatges.

```
GraphicsWindow.BackgroundColor = "blue"  
GraphicsWindow.Title = "El meu gràfic"  
GraphicsWindow.Width = 500  
GraphicsWindow.Height = 400  
GraphicsWindow.Show()
```

- ' Canvi de color de fons
- ' Títol de la finestra
- ' Canvi d'amplada de la finestra
- ' Canvi d'alçada de la finestra

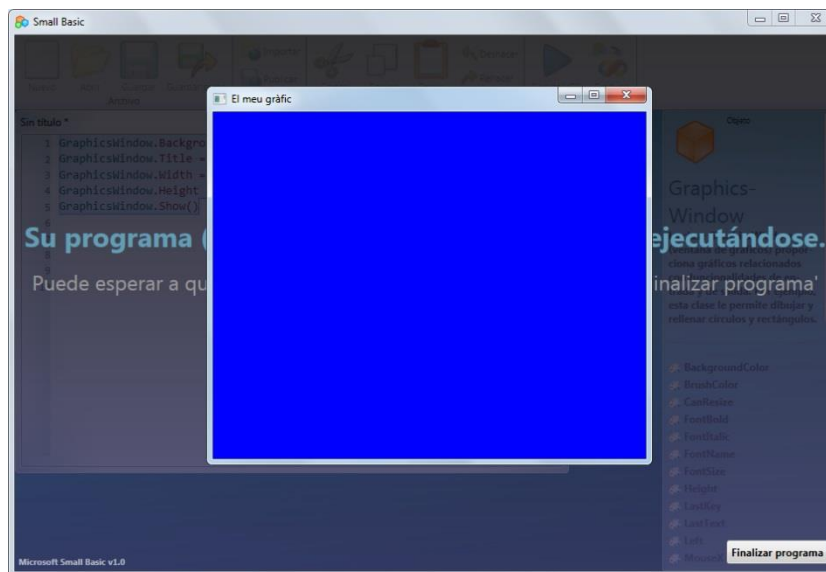


Figura 5 – Configuració finestra gràfica

## Dibuixem línies

```
GraphicsWindow.Width = 400
GraphicsWindow.Height = 400
GraphicsWindow.DrawLine(20, 10, 200, 10)
GraphicsWindow.DrawLine(20, 40, 200, 40)
```

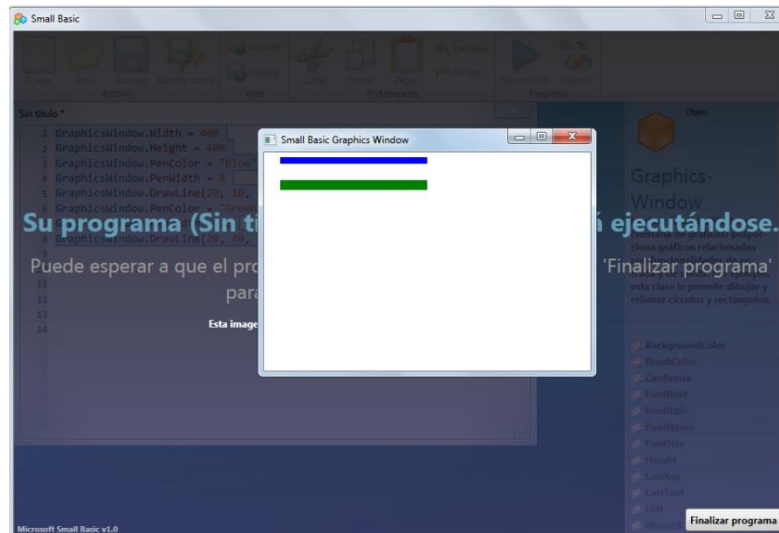
Aquest exemple dibuixa dues línies paral·leles. La instrucció utilitzada *DRAWLINE* requereix de quatre coordenades (x1,y1,x2,y2). Les primeres corresponen a la coordenada (x1,y1) inicial d'on comença la línia i les següents corresponen a la coordenada (x2,y2) d'on finalitza la mateixa línia.

Les coordenades de les finestres gràfiques comencen a la cantonada superior esquerra de la finestra (0,0).

El mode gràfic de Small Basic ens permet canviar l'amplada de les línies i el seu color. A l'exemple anterior posarem a cada línia una amplada i color diferent.

```
GraphicsWindow.Width = 400
GraphicsWindow.Height = 400
GraphicsWindow.PenColor = "Blue"
GraphicsWindow.PenWidth = 8
GraphicsWindow.DrawLine(20, 10, 200, 10)
GraphicsWindow.PenColor = "Green"
GraphicsWindow.PenWidth = 12
GraphicsWindow.DrawLine(20, 40, 200, 40)
```

- ' Canviem color de la línia
- ' Canviem amplada de la línia



**Figura 6 – Dibuixem i modifiquem línies**

Aprofitem ara que a sessions anteriors es van estudiar les estructures repetitives. Per tant farem un exemple per dibuixar 10 línies que van canviant la seva amplada automàticament.

```
GraphicsWindow.BackgroundColor = "White"
GraphicsWindow.Width = 400
GraphicsWindow.Height = 260
GraphicsWindow.PenColor = "Green"

For i = 1 To 10
    GraphicsWindow.PenWidth = i
    GraphicsWindow.DrawLine(20, i * 15, 280, i * 15)
EndFor
```

### **Dibuixem formes**

Small Basic ens permet moltes formes geomètriques conegudes, com rectangles, quadrats, triangles etc. A més a més, podem canviar el color i tamany de la ploma que dibuixa la vora i el color de fons.

L'exemple següent ens presentarà dos rectangles, un amb la vora de color blau i sense color de fons, i l'altre el pintarem amb color groc de fons i sense vora.

```
GraphicsWindow.Width = 500
GraphicsWindow.Height = 400
GraphicsWindow.PenColor = "blue"
GraphicsWindow.BrushColor = "yellow"
GraphicsWindow.DrawRectangle(20, 20, 100, 300)
GraphicsWindow.FillRectangle(200, 20, 100, 300)
```

- ' Color de la vora
- ' Color de fons
- ' Dibuixa la forma sense fons
- ' Dibuixa la forma amb fons



Les funcions *DrawRectangle* i *FillRectangle* utilitzen 4 coordenades. Les dues primeres són el punt (x,y) de la finestra on començarem a dibuixar la figura i les dues següents són l'amplada i l'alçada de la forma que volem dibuixar. Per tant, si volem dibuixar un quadrat només haurem de modificar les coordenades del rectangle per convertir-lo a quadrat.

```
GraphicsWindow.Width = 500
GraphicsWindow.Height = 400
GraphicsWindow.PenColor = "blue"
GraphicsWindow.BrushColor = "yellow"
GraphicsWindow.DrawRectangle(20, 20, 100, 100)
GraphicsWindow.FillRectangle(200, 20, 100, 100)
```

Ara dibuixarem una el·lipse i un cercle utilitzant la funció *DrawEllipse*.

```
GraphicsWindow.Width = 500
GraphicsWindow.Height = 400
GraphicsWindow.PenColor = "blue"
GraphicsWindow.BrushColor = "yellow"
GraphicsWindow.DrawEllipse(20, 20, 200, 100)
GraphicsWindow.FillEllipse(300, 20, 100, 100)
```

Com es pot observar quan s'executa el programa, primer es dibuixa una el·lipse i després un cercle, però hem utilitzat la mateixa funció. Això vol dir que per obtenir un cercle utilitzem la funció el·lipse modificant l'amplada i alçada.

### Combinació de formes i estructures

Ara farem un exemple on combinarem una estructura repetitiva i la funció que dibuixa cercles per obtenir una figura molt atractiva.

```
GraphicsWindow.BackgroundColor = "Black"
GraphicsWindow.PenColor = "Blue"
GraphicsWindow.Width = 200
GraphicsWindow.Height = 200

For i = 1 To 100 Step 5
    GraphicsWindow.DrawEllipse(100 - i, 100 - i, i * 2, i * 2)
EndFor
```

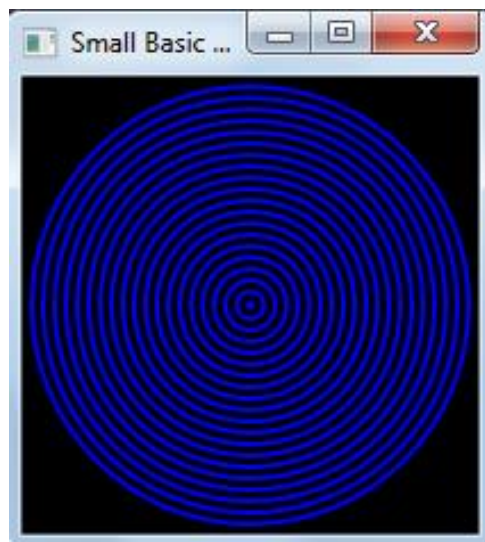


Figura 5 – Dibuixem formes

Un altre exemple és com el de la figura 6, que fa servir funcions matemàtiques aleatòries per obtenir un espectacular mosaic de cercles de diferents colors. La funció *GetRandomColor()* ens proporcionarà de forma aleatòria un color de tots els que fa servir Small Basic i la funció *GetRandomNumber()* ens donarà un número aleatori que farem servir per les coordenades de posicionament del cercle que volem dibuixar.

```
GraphicsWindow.BackgroundColor = "Black"
For i = 1 To 500
  GraphicsWindow.BrushColor = GraphicsWindow.GetRandomColor()
  x = Math.GetRandomNumber(640)
  y = Math.GetRandomNumber(480)
  GraphicsWindow.FillEllipse(x, y, 20, 20)
EndFor
```

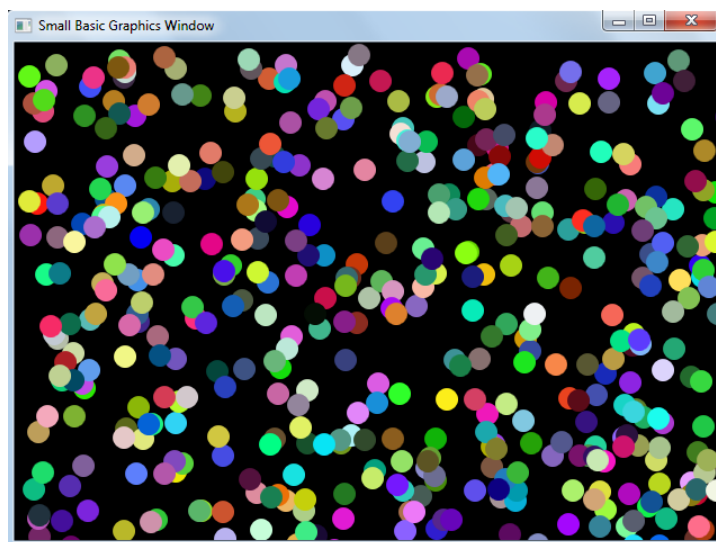


Figura 6 – Dibuixem formes aleatòries

### Proposta d'exercici:

Escriu un programa en SMALL BASIC que dibuixi dues figures. Una de cercles repetitius de diferents colors i l'altre quadrats repetitius també de diferents colors.

### Solució:

```
GraphicsWindow.BackgroundColor = "Black"
GraphicsWindow.Width = 500
GraphicsWindow.Height = 500

For i = 1 To 100 Step 5
    GraphicsWindow.PenColor = GraphicsWindow.GetRandomColor()
    GraphicsWindow.DrawEllipse(100 - i, 100 - i, i * 2, i * 2)
EndFor

For i = 1 To 100 Step 5
    GraphicsWindow.PenColor = GraphicsWindow.GetRandomColor()
    GraphicsWindow.DrawRectangle(300 - i, 300 - i, i * 2, i * 2)
EndFor
```

## Sessió 4. Introducció II. El mode gràfic a SMALL BASIC 2ª Part.

### Nota:

Aquesta sessió permetrà introduir nous conceptes de l'entorn gràfic de Small Basic. Introduïrem el concepte de la tortuga gràfica que ens permetrà dibuixar qualsevol figura que ens vingui en ment i aprendrem tot sobre els esdeveniments i les subrutines, que són les formes de respondre a una acció concreta que fem amb el ratolí.

- ✓ Es pot dir que la resposta a esdeveniments és la forma més interactiva de programar ja que la gran majoria de pàgines web i d'aplicacions conegudes fan servir el control dels esdeveniments de forma molt habitual.

### La tortuga gràfica

Small Basic incorpora l'anomenada tortuga gràfica que obeeix a diferents ordres de moviment mitjançant instruccions. Amb aquestes ordres podem dibuixar qualsevol figura que li demanem.

El primer que hem de fer és mostrar la tortuga a la finestra gràfica. Per això utilitzarem la següent instrucció:

```
Turtle.Show()
```

Una de les primeres ordres que li podem donar a la tortuga és que es mogui mitjançant la instrucció Small Basic:

```
Turtle.Move(200)
```

Amb això podem observar com la tortuga es mou lentament 200 posicions o píxels per la pantalla. Per tant, amb la combinació de diferents instruccions que li podem donar a la tortuga podrem dibuixar moltes figures.

Per exemple, si volem dibuixar un quadrat de 100 píxels de costat ho farem de la següent forma:

```
Turtle.Move(100)
Turtle.TurnRight()
Turtle.Move(100)
Turtle.TurnRight()
Turtle.Move(100)
Turtle.TurnRight()
Turtle.Move(100)
Turtle.TurnRight()
```

Com es pot veure estem repetint quatre instruccions per fer el quadrat. En sessions anteriors vam aprendre les estructures repetitives, per tant modificarem el programa anterior per fer-lo amb una estructura repetitiva.

```
For i = 1 To 4
    Turtle.Move(100)
    Turtle.TurnRight()
EndFor
```

Si volem que la tortuga tingui més velocitat només haurem d'afegir la instrucció següent:

```
Turtle.Speed = 10
```

Ara dibuixarem el mateix quadre anterior però posant un color diferent a cada costat.

```
For i = 1 To 4
    GraphicsWindow.PenColor = GraphicsWindow.GetRandomColor()
    Turtle.Move(100)
    Turtle.TurnRight()
EndFor
```

Small Basic disposa d'una instrucció *Turn()* que permet girar la tortuga l'angle que especifiquem dins la instrucció. Per exemple, si volem dibuixar un pentàgon haurem de dibuixar cinc costats iguals i per tant haurem de saber l'angle de gir. Si  $360^\circ$  graus ho dividim per 5 costats obtindrem l'angle per fer girar la tortuga.  $360/5 = 72^\circ$

```
For i = 1 To 5
    Turtle.Move(100)
    Turtle.Turn(72)
EndFor
```

Ara observem que passa amb el següent programa:

```
costats = 10

llarg = 400 / lados
angle = 360 / lados

Turtle.Speed = 10

For i = 1 To costats
    Turtle.Move(llarg)
    Turtle.Turn(angle)
EndFor
```

Aquest programa ens permet modificant diferents paràmetres dibuixar quasi qualsevol figura. Per exemple si volem dibuixar un cercle només caldrà afegir més quantitat al paràmetre costats. Si posem valor 100 a costats es podrà observar com la tortuga dibuixa un cercle.

La tortuga també es pot moure sense que escrigui res en pantalla. Per això utilitzarem la instrucció *Turtle.PenUp()* per que no dibuixi i *Turtle.PenDown()* per què torni a dibuixar.

Amb aquesta combinació i l'exemple anterior podem dibuixar per exemple un cercle amb línies discontinues:

```
costats = 50

llarg = 400 / costats / 2
angle = 360 / costats / 2

Turtle.Speed = 10

For i = 1 To costats
    Turtle.PenDown()
    Turtle.Move(llarg)
    Turtle.Turn(angle)
    Turtle.PenUp()
    Turtle.Move(llarg)
    Turtle.Turn(angle)
EndFor
```

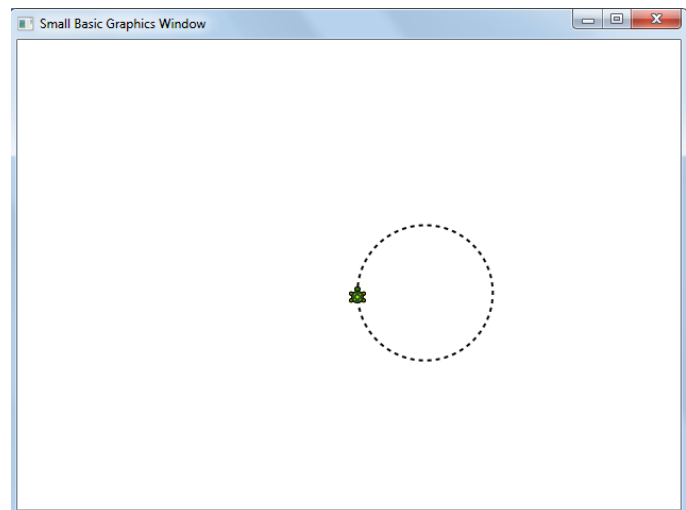
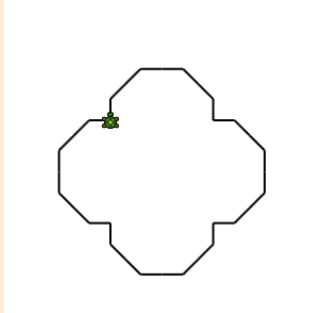


Figura 7 – Dibuixant un cercle discontinuo

### Proposta d'exercici:

Escriu un programa en SMALL BASIC que dibuixi una figura com aquesta amb la tortuga utilitzant estructures repetitives:



### Solució:

```
For j = 1 To 4
  For i = 1 To 4
    Turtle.PenDown()
    Turtle.Move(20)
    Turtle.Turn(45)
    Turtle.Move(20)
  EndFor
  Turtle.PenUp()
  Turtle.Turn(-90)
  Turtle.PenDown()
EndFor
```

### Els eventos

Els eventos són indispensables per introduir interactivitat a qualsevol programa. Per poder explicar la interactivitat en Small Basic cal explicar prèviament que són les subrutines.

Les subrutines són una part del codi dins del programa que habitualment fa una funció específica i que es pot cridar des de qualsevol part del programa. Les subrutines estan identificades per un nom que segueix a la paraula clau *Sub* i acaba amb la paraula clau *EndSub*.

A continuació posem un exemple de com es fa servir una subrutina. Aquest exemple crida una subrutina dues vegades.

```
ImprimirHora()
TextWindow.Write("Com et dius: ")
nom = TextWindow.Read()
TextWindow.Write(nom + ", la hora és: ")
ImprimirHora()

Sub ImprimirHora
  TextWindow.WriteLine(Clock.Time)
EndSub
```

Ara que ja sabem com funcionen les subrutines, les utilitzarem per explicar els eventes. Un primer exemple d'evente és que la pantalla ens presenti una petita finestra informativa quan polsem un botó del ratolí. Per fer això, farem servir l'evente *MouseDown* que al activar-se mitjançant el botó del ratolí cridarà la subrutina que ens ensenyarà una finestra informativa.

```
GraphicsWindow.MouseDown = botoMouse

Sub botomouse
GraphicsWindow.ShowMessage("Has polsat el botó del ratolí", "Boto ratolí")
EndSub
```

El següent exemple dibuixa un quadrat vermell quan es polsa el botó del ratolí a la posició exacte on està situat el punter, cridant la subrutina que capta la posició del ratolí amb les comandes *MouseX* i *MouseY*.

```
GraphicsWindow.BrushColor = "Red"
GraphicsWindow.MouseDown = BotoRatoli

Sub BotoRatoli
x = GraphicsWindow.MouseX - 10
y = GraphicsWindow.MouseY - 10
GraphicsWindow.FillRectangle(x, y, 20, 20)
EndSub
```

### Proposta d'exercici:

Escriu un programa en SMALL BASIC que dibuixi cercles blaus quan es mogui el ratolí.

### Solució:

```
GraphicsWindow.MouseMove = BotoEsMou
GraphicsWindow.BrushColor = "Blue"

Sub BotoEsMou
x = GraphicsWindow.MouseX - 10
y = GraphicsWindow.MouseY - 10
GraphicsWindow.FillEllipse(x, y, 20, 20)
EndSub
```

## Sessió 5 i 6. Introducció II. Activitat Final.

Aquestes dues sessions es dedicarien a preparar un petit projecte en grup per a que els alumnes programin la següent solució.

### Proposta d'Activitat Final:

Fer l'algoritme i el programa en SMALL BASIC que faci la comparació de diferents costos de telèfons segons la tarifa escollida d'un proveïdor i calculi el cost final que tindria el telèfon després de pagar les quotes de durada del contracte escollit.

### Possible solució:

```
TextWindow.Write("Número de contractes (min = 2, max = 4) que vols  
comparar: ")  
numcontracts = TextWindow.Read()  
  
While numcontracts < 2 Or numcontracts > 4  
TextWindow.Write("ERROR D'ENTRADA! El número de contractes ha de ser entre  
2 i 4, torna a entrar:")  
numcontracts = TextWindow.Read()  
EndWhile  
  
TextWindow.WriteLine("")  
TextWindow.WriteLine("TIPUS DE CONTRACTE")  
TextWindow.WriteLine("")  
TextWindow.WriteLine("1: 12 Mesos")  
TextWindow.WriteLine("2: 18 Mesos")  
TextWindow.WriteLine("3: 24 Mesos")  
TextWindow.WriteLine("")  
  
TextWindow.Write("Si us plau entra el tipus de contracte: 1, 2 o 3: ")  
option = TextWindow.Read()  
  
While option < 1 Or option > 3  
TextWindow.Write("ERROR! Entra la opció correcta 1, 2 o 3: ")  
option = TextWindow.Read()  
EndWhile  
  
If option = 1 Then  
contractlength = 12  
Else  
If option = 2 Then  
contractlength = 18  
  
Else  
contractlength = 24  
EndIf  
EndIf
```



```

For counter = 1 To numcontracts
    TextWindow.WriteLine("")
    TextWindow.Write ("Entra el nom del fabricant del telèfon: ")
    manufacturer[counter] = TextWindow.Read()
    TextWindow.Write ("Entra el model del telèfon: ")
    model[counter]= TextWindow.Read()
    TextWindow.Write ("Entra el cost inicial del telèfon: ")
    initialcost[counter]= TextWindow.Read()
    TextWindow.Write ("Entra els cost mensual del contracte: ")
    costpermonth[counter]= TextWindow.Read()
EndFor

For totalcounter = 1 To numcontracts
    totalcost[totalcounter] = (contractlength * costpermonth[totalcounter])
+ initialcost[totalcounter]
EndFor

TextWindow.Clear()
TextWindow.ForegroundColor = "White"
TextWindow.WriteLine ("COMPARACIÓ DE TELÈFONS ")
TextWindow.WriteLine ("Contracte escollit: " + contractlength + " mesos.")
TextWindow.WriteLine("")
TextWindow.Write ("Fabricant")
TextWindow.CursorLeft = 15
TextWindow.Write ("Model")
TextWindow.CursorLeft = 25
TextWindow.Write ("Cost Inicial")
TextWindow.CursorLeft = 40
TextWindow.Write ("Cost Per Mes")
TextWindow.CursorLeft = 60
TextWindow.Write ("Total Cost")
TextWindow.WriteLine("")

For counter2 = 1 To numcontracts
    TextWindow.Write(manufacturer[counter2])
    TextWindow.CursorLeft = 15
    TextWindow.Write(model[counter2])
    TextWindow.CursorLeft = 25
    TextWindow.Write( initialcost[counter2] + " Eur" )
    TextWindow.CursorLeft = 40
    TextWindow.Write(costpermonth[counter2] + " Eur")
    TextWindow.CursorLeft = 60
    TextWindow.WriteLine(totalcost[counter2] + " Eur")
EndFor

TextWindow.WriteLine("")

```

## 8. CONCLUSIONS

Aquest material didàctic que he elaborat és fàcil i econòmic d'implantar a qualsevol institut. És imprescindible que hi hagi un aula equipada amb ordinadors i que tinguin instal·lat l'entorn Small Basic que es pot aconseguir gratuïtament de la pàgina web del seu fabricant Microsoft Corporation.

Crec que els objectius plantejats a les diferents sessions dissenyades proporcionen al professor un material adequat que pot servir d'ajuda en la seva funció d'introduir els alumnes a aprendre programació d'ordinadors amb una barreja de conceptes teòrics que tenen aplicació pràctica a quasi totes les sessions.

Pel que fa als objectius generals de l'elaboració d'aquest TFM que m'havia plantejat a l'inici puc concloure que:

*Objectiu 1. DONAR resposta a la inquietud generada pel responsable de la matèria d'Informàtica a 4rt d'ESO del centre on vaig fer el pràcticum per tenir un material per treballar a l'aula:*

**Conclusió 1: Amb la redacció d'aquest TFM he pogut donar resposta al professor d'Informàtica de 4rt d'ESO que volia tenir un material per posar en pràctica amb els seus futurs alumnes del curs 2014-15.**

*Objectiu 2. PROPOSAR activitats en format de material didàctic per a poder dur a terme a l'aula d'informàtica de qualsevol centre:*

**Conclusió 2: Aquest TFM compleix tot els requisits per ser utilitzat a qualsevol centre del nostre país ja que no es necessari cap recurs material, a banda d'una aula d'informàtica, ni cap recurs econòmic ja que el programari utilitzat és totalment gratuït.**

*Objectiu 3. ADJUNTAR una pauta de recomanacions i solucions a les activitats per professorat no expert en programació:*

**Conclusió 3: Amb la inclusió de notes d'ajuda i la resolució de les activitats plantejades crec que professor no expert en programació podria utilitzar el material sense cap mena de complicació.**

*Objectiu 4. POSAR a prova, jo mateix, aquest material amb els alumnes del centre on feia el pràcticum:*

**Conclusió 4: És molt difícil posar a prova el material didàctic que he elaborat en aquest TFM ja que la programació de les diferents matèries d'un Institut està feta a l'inici de curs, i aquest material no s'acaba fins els mes de març o abril.**

Particularment vaig intentar posar-lo a prova els últims dies de la meva estada al centre de pràctiques, però no vaig poder ja que el professor de la matèria ja va programar anteriorment a la meva arribada una sèrie d'activitats on era impossible encabir 12 sessions d'aquest projecte. Em van oferir utilitzar 2 sessions però vaig considerar que era impossible resumir tot en tant poc temps i quedaria massa incomplet de cara a la comprensió dels propis alumnes.

Resumint, puc concloure que la programació d'ordinadors és una eina educativa que introdueix el pensament computacional als nostres alumnes i que és la base de moltes professions que actualment requereixen d'aquest tipus de pensament o futures professions que seran indispensables per a competir a la societat del futur. Per això, crec que hem d'ensenyar a programar als nostres alumnes que seran els futurs professionals de la societat.

## 9. BIBLIOGRAFIA I REFERÈNCIES

[1] M.A. Fernández. Metodología de la programación a través de pseudocódigo.1ª Edició. McGraw Hill. 1991. ISBN 84-7615-633-2.[en suport paper].

[2] Microsoft Corporation. SMALL BASIC Tutorial. An Introduction to programming. 2013 [en línia].[Consulta: 10 de maig de 2014]. Disponible a: < <http://smallbasic.com> >

[3] Generalitat de Catalunya Departament d'Ensenyament. XTEC. Xarxa Educativa Telemàtica. Desplegament del Currículum de l'ESO. 2007. [en línia].[Consulta: 3 de juny de 2014]. Disponible a: <[http://www.xtec.cat/alfresco/d/d/workspace/SpacesStore/fe124c3b-2632-44ff-ac26-dfe3f8c14b45/curriculum\\_eso.pdf](http://www.xtec.cat/alfresco/d/d/workspace/SpacesStore/fe124c3b-2632-44ff-ac26-dfe3f8c14b45/curriculum_eso.pdf)>

[4] UNESCO. Informatique pour l'enseignement secondaire.1996.[en línia].[Consulta: 5 de maig de 2014]. Disponible a: < <http://unesdoc.unesco.org/images/0009/000973/097323fb.pdf>>

[5] Murphy, S. Finland Eyes Programming Classes for Elementary School Students. 2013. [en línia].[Consulta: 2 de mayo de 2014]. Disponible a: <<http://mashable.com/2013/11/16/finland-tech-education-schools/>>

[6] Fundación Tecnológica HITSA. Plan Estonio ProgeTiiger. 2012. [en línia].[Consulta: 5 de maig de 2014]. Disponible a: < <http://www.innovatsioonikeskus.ee/en/programming-schools-and-hobby-clubs> >

[7] Resnick, M. Let's teach kids to code. 2012. [en línia].[Consulta: 10 de maig de 2014]. Disponible a: < [http://www.ted.com/talks/mitch\\_resnick\\_let\\_s\\_teach\\_kids\\_to\\_code](http://www.ted.com/talks/mitch_resnick_let_s_teach_kids_to_code) >

[8] Coughlan, S. Computer science part of English Baccalaureate. 2013. [en línia].[Consulta: 5 de maig de 2014]. Disponible a: < <http://www.bbc.com/news/education-21261442> >

[9] Petit, A. Comment développer la culture en informatique : en l'enseignant dès le lycée. 2010. [en línia].[Consulta:10 de maig de 2014]. Disponible a: <[https://interstices.info/cms/i\\_57558/comment-developper-la-culture-en-informatique-en-l-enseignant-des-le-lycee](https://interstices.info/cms/i_57558/comment-developper-la-culture-en-informatique-en-l-enseignant-des-le-lycee)>